Newton’s Polynomial: **General Form (nth order)**

- <u>Note:</u> Calculating these divided differences is a **recursive** process:

$$f_{dd}[x_3, x_2, x_1, x_0]$$

$$f_{dd}[x_2, x_1, x_0] \quad f_{dd}[x_3, x_2, x_1]$$

$$f_{dd}[x_1, x_0] \qquad f_{dd}[x_2, x_1] \qquad f_{dd}[x_3, x_2]$$

$$f(x_0) \qquad f(x_1) \qquad f(x_2) \qquad f(x_3)$$

Newton's Polynomial: **General Form (n^{th} order)**

- Note: Calculating these divided differences is a **recursive** process:

$$f_{dd}[x_3, x_2, x_1, x_0]$$

$$f_{dd}[x_2, x_1, x_0] \quad f_{dd}[x_3, x_2, x_1]$$

$$f_{dd}[x_1, x_0] \qquad f_{dd}[x_2, x_1] \qquad f_{dd}[x_3, x_2]$$

$$f(x_0) \qquad f(x_1) \qquad f(x_2) \qquad f(x_3)$$

Newton’s Polynomial: **General Form (n^th order)**

- <u>Note:</u> Calculating these divided differences is a **recursive** process:

$$f_{dd}[x_3, x_2, x_1, x_0]$$

Store these for later use!

$$f_{dd}[x_2, x_1, x_0] \qquad f_{dd}[x_3, x_2, x_1]$$

$$f_{dd}[x_1, x_0] \qquad f_{dd}[x_2, x_1] \qquad f_{dd}[x_3, x_2]$$

$$f(x_0) \qquad f(x_1) \qquad f(x_2) \qquad f(x_3)$$

Newton's Polynomial: **General Form (nth order)**

- Putting these divided differences into the polynomial form:

$$
\begin{aligned}
f_3(x) = {} & f(x_0) \\
& + f_{dd}[x_1, x_0](x - x_0) \\
& + f_{dd}[x_2, x_1, x_0](x - x_0)(x - x_1) \\
& + f_{dd}[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

Newton’s Polynomial: **General Form (nth order)**

- Putting these divided differences into the polynomial form:

$$
\begin{aligned}
f_n(x) = f(x_0) \\
&+ f_{dd}[x_1, x_0](x - x_0) \\
&+ f_{dd}[x_2, x_1, x_0](x - x_0)(x - x_1) \\
&+ f_{dd}[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2) \\
&\quad \vdots \qquad\qquad\qquad\qquad \vdots \\
&+ f_{dd}[x_n, x_{n-1}, \dots, x_0](x - x_0)(x - x_1) \cdots (x - x_{n-1})
\end{aligned}
$$

- This is the general form of Newton’s divided-difference interpolating polynomial

Newton's Polynomial: **Example 18.1 & 18.2 – ln(x)**

- Estimate $f(2)$ using 3rd order polynomial:

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + b_3(x - x_0)(x - x_1)(x - x_2)$$

- Need 4 points, so add one more

	x	$f(x)$
x_0	1	0
x_1	4	1.386294
x_2	6	1.791759
x_3	5	1.609438

Newton's Polynomial: **Example 18.1 & 18.2 – ln(x)**

- Estimate $f(2)$ using 3rd order polynomial:

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$
$$+ b_3(x - x_0)(x - x_1)(x - x_2)$$

	x	$f(x)$
x_0	1	0
x_1	4	1.386294
x_2	6	1.791759
x_3	5	1.609438

- Need 4 points, so add one more
  - Note that points need be sequential

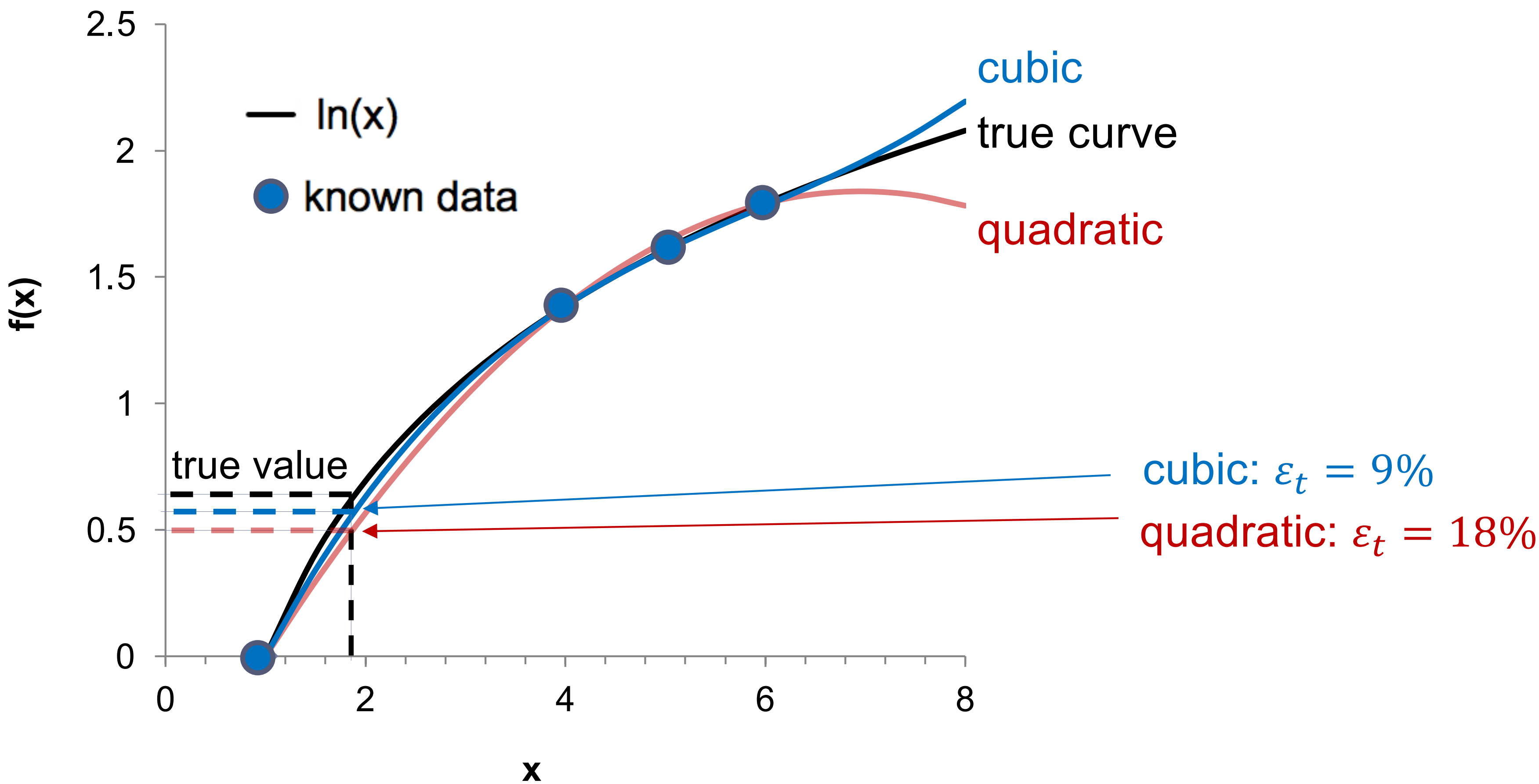
- Recall:

$$b_0 = f(x_0)$$
$$b_1 = f_{dd}[x_1, x_0]$$
$$b_2 = f_{dd}[x_2, x_1, x_0]$$
$$b_3 = f_{dd}[x_3, x_2, x_1, x_0]$$

MIE334_Lecture_21_ExNewt3.xlsx

Newton's Polynomial: **Example 18.3 - Results**

- Higher order further reduces the true error



Newton’s Polynomial: **Error**

- General form of this polynomial looks like a Taylor series:

$$
\begin{aligned}
f_n(x) = f(x_0) & \\
& + f_{dd}[x_1, x_0](x - x_0) \\
& + f_{dd}[x_2, x_1, x_0](x - x_0)(x - x_1) \\
& \quad \vdots \qquad\qquad\qquad \vdots \\
& + f_{dd}[x_n, x_{n-1}, \ldots, x_0](x - x_0)(x - x_1) \cdots (x - x_{n-1})
\end{aligned}
$$

- Successive terms are added to estimate higher-order behavior (curvature) of the function

Newton's Polynomial: **Error**

- We can derive an estimate of the error in our interpolation using a property of the Taylor series

- Recall: Truncation error for the Taylor series:
  - $R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(\overbrace{x_{i+1} - x_i}^{h})^{n+1}$
    - ξ is somewhere (unknown) between x_{i+1} and x_i

- For a n^th-order polynomial interpolation we can use an analogous term to replace the $(x_{i+1} - x_i)^{n+1}$ term:
  - $(x - x_0)(x - x_1) \ldots (x - x_n)$
    - Still a polynomial of order $n + 1$

Newton's Polynomial: **Error**

- So error term now looks like:
  - $R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \dots (x - x_n)$
    - ξ lies somewhere in range of data points

- We can estimate the (n+1) derivative using a (n+1) finite divided difference:
  - $\frac{f^{(n+1)}(\xi)}{(n+1)!} \cong f_{dd}[x_{n+1}, x_n, x_{n-1}, \dots, x_0]$

- To estimate the error, we need one additional data point:
  - $R_n = f_{dd}[x_{n+1}, x_n, x_{n-1}, \dots, x_0](x - x_0)(x - x_1) \dots (x - x_n)$

Newton's Polynomial: Error - Example

- Earlier, we estimated f(2) with a 2nd-order polynomial
  - Original data points are the same
  - But use additional point, $f(x_3) = f(5)$, to estimate the error for this interpolation

x	$f(x)$
1	0
4	1.386294
6	1.791759
5	1.609438

- From the 2nd order fit we found:
  - $f_2(2) = 0.565844, \;\; \varepsilon_t = 18.4\%$

- $R_2 = f_{dd}[x_3, x_2, x_1, x_0\,](x - x_0)(x - x_1)(x - x_2)$
  $= f_{dd}[5,6,4,1](2 - 1)(2 - 4)(2 - 6) = \mathbf{0.062924}$
  - Same order of magnitude as true error ($E_t = 0.127303$) but about half the value